U.S. Department of Energy

Office of Management, Budget and Evaluation

Systems Engineering and Interface Management

Initiated by: Office of Engineering and Construction Management

SYSTEMS ENGINEERING AND INTERFACE MANAGEMENT

The primary goal of the systems engineering process is to transform mission operational requirements or remediation requirements into system architecture, performance parameters, and design details. Beginning with the definition of a need, systems engineering is a process that progresses through the establishment of functions and requirements, performance of functional analyses, the identification and evaluation of alternatives, the solution of a preferred alternative, and validation of the preferred alternative. The process ends with verification that the need is met, including interfaces, fit, and completeness. The application of systems engineering to a project is tailored to the project's needs.

1.0 SYSTEMS ENGINEERING OVERVIEW

This section addresses the basic systems engineering process for projects. The primary goal of the systems engineering process is the transformation of mission, operational, or disposal requirements into a system architecture, performance parameters, and design details.

1.1 Systems Engineering Principles

The systems engineering process can be viewed as a hierarchy beginning with the definition of a need, progressing through to a baseline and ending with verification that the need has been met. The principles of the hierarchy are:

- Need—what is the need for this project?
- Functions—what are the functions to be performed by this project?
- Requirements—what are the performance requirements and constraints on the project?
- Criteria—what criteria shall be used to select among alternatives for performing the functions and meeting the requirements?
- Alternatives—what alternative solutions are available to perform the functions and meeting the requirements?
- Technical Baseline—what is the best or the preferred alternative for performing the functions and meeting the requirements?
- Verification—what is the proof that the preferred alternative performs the functions and meets the requirements?

The process of establishing functions and requirements is generally straightforward for low risk, simple projects. On the other hand, complex systems require extensive processes to identify all applicable requirements. The identification of complete sets of functions and requirements necessarily involves iteration. The iterative process will yield information that should be included in project specifications, and the specifications should baseline the requirements.

The process involves iterative applications of a series of steps: mission analysis, function definitions, requirements identification, architecture development, alternative identification, engineering tradeoff studies based on prepared decision criteria, and test and evaluation. The starting point for the systems engineering process is generally the high-level mission statement followed by each of the listed steps. This establishes the Level-1 baseline that is followed by another iteration performing the same steps at the next level of detail. The next level of detail may be based on logical allocation of the mission statement, functions, or architecture. This establishes the start point for developing the Level-2 baseline as the iteration continues.

At each subsequent level, the functions from the previous level are evaluated and subdivided (allocated) to identify all the sub-functions that are necessary and sufficient to accomplish the parent function (previous level). For example, a parent function to “remediate waste” might be decomposed into three sub-functions of “retrieve waste,” “process waste,” and “store waste.” After completing the functional analysis at a new level, requirements from the previous level are evaluated and allocated to the sub-functions and additional requirements identified. The remaining steps of the systems engineering process have to be performed at each level before going to the next level.

Once the requirements and functions have been established, an essential step in the systems engineering process is the broad search for viable alternatives that will perform the functions and meet the requirements. The systems engineering process encourages creativity and innovation in the search for end products to perform functions and requirements by explicitly identifying the search for alternatives. On the other hand, the systems engineering process places great emphasis on testing and quantitative data to demonstrate that alternatives, which may be very creative and innovative, are a viable means for performing the functions and meeting the requirements. The search for alternatives should involve iteration with the functions and requirements steps of the systems engineering process.

Selection among the viable alternatives should be based on predetermined criteria. The mission analysis should identify goals, objectives, and values that help determine the selection criteria. In most cases, the criteria are obvious considerations such as cost, time, and risk. Increasingly, however, less obvious criteria such as organizational or stakeholder values need to be considered in the selection process.

Preferred alternatives resulting from the selection process become the technical baseline for the project. Once the technical baseline has been selected, it becomes the basis for control of the iterative processes. Iterative refinement of this technical baseline adds details and eventually yields the end product that has been verified to perform the functions and meet the requirements.

When and where to stop the iterative processes is a key question. High-level mission statements, such as site cleanup usually stop their iteration with the identification of systems such as a preprocessing plant or a high-level waste vitrification plant. Each system then may be handed off to another organization that continues the iteration process down to the design requirements for each system, using the same basic steps.

Some indications of when the iteration process should be stopped include:

- When a well-bounded end product has been identified
- When no more practical functional allocations exist
- When the end product can be provided using existing technology
- When the end product is affordable
- When, for the level of allocation, sufficient information is available to make the required decisions for the next set of activities (i.e., continue to the next phase or stop project work)
- When the complexity and quantity of data have reached a point that one organization cannot manage the information effectively (i.e., discrete systems are broken out to be worked by different organizations)
- When an organization has allocated to a level at which it performs a make-buy analysis. If a make decision is made,the requiremetns are included in specifications and drawings. If a buy decision is made, the requirements are included in a contract.

Program and project managers are responsible for implementing a systems engineering process when planning a project. The planning activity should consider the following factors:

- Experience and skills mix of the project staff
- Documentation of the engineering methodology and tools selected
- Staff access to a senior technical expert in the subject of the end product
- How the selected approach yields an end product that meets the customer’s needs.

1.2 Systems Engineering Process

The systems engineering process shown in Figure 1 consists of four activities that are performed in a logical sequence and supported by three additional concurrent engineering processes. This process is described in detail in the following sections.

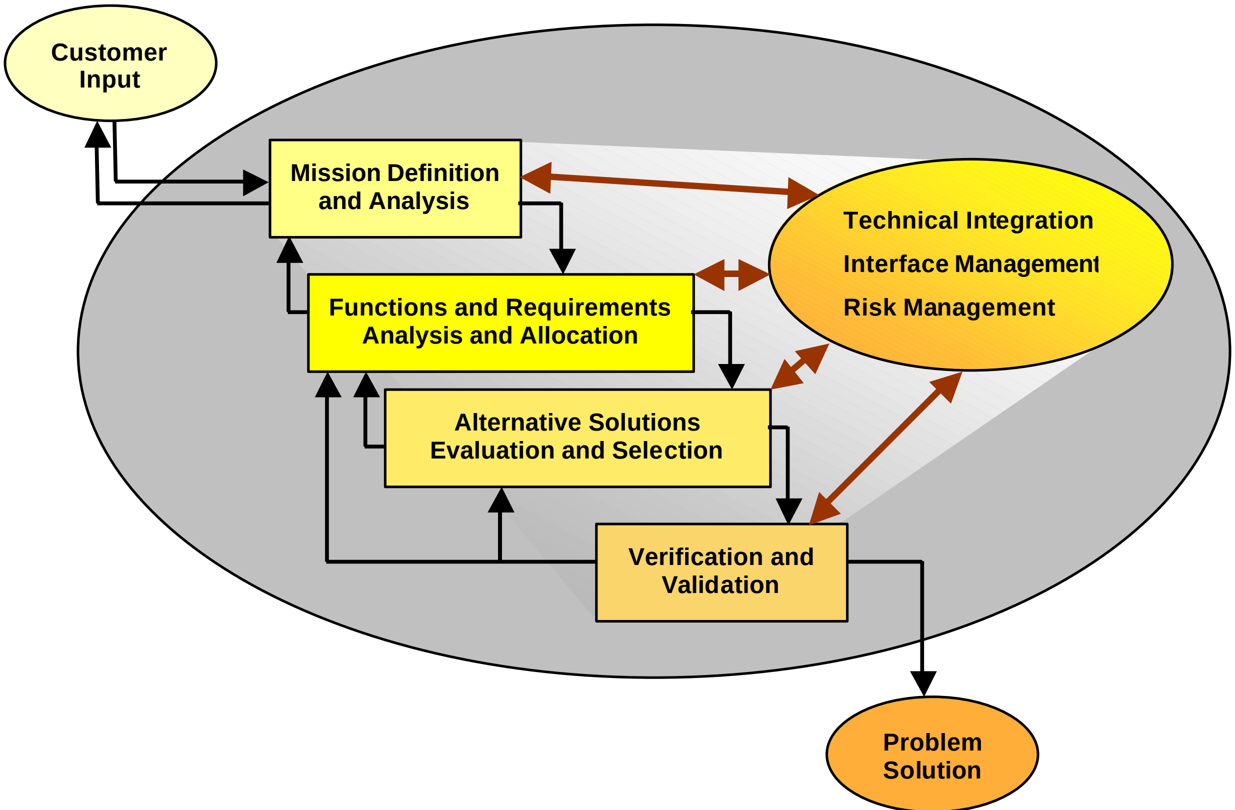

Figure 1. Systems Engineering Process

2.0 INPUTS TO SYSTEMS ENGINEERING

Defining functions, identifying requirements and identifying appropriate systems engineering architecture are three vital steps/activities in the systems engineering process. These steps/activities are described in the following paragraphs.

2.1 Functions

Determining a project's functions is one of the key steps supporting project planning and project definition. Accurate formulation of the functions leads to assurance that the end product will satisfy the mission need. Once defined, the functions are the basis for many other planning elements including design criteria, specifications, work breakdown structure, cost estimates, alternative evaluations, and value management studies.

A function is a task, action, or activity that must be performed. A function receives an input, does something to it, and produces an output. It may require one or more inputs and may produce one or more outputs. A function is generally described using a verb/noun combination such as "remediate waste" and has an interface with at least one other function. A function describes what must be done, not how.

The initial, high-level project functions are established during pre-acquisition activities. However, they are not normally of sufficient detail to produce the end product. Additional detail is developed during the Conceptual and Execution phases through an allocation process.

Functional allocation in conjunction with the other steps of the systems engineering process is a top down approach for developing a functional definition of the end product and the translation of high-level functions into detailed design criteria. It is an iterative process of breaking functions down from the system level to the subsystem and as far down as necessary to identify detailed design criteria for the various components of the system.

Those performing the functional allocation need to ensure that each function has at least one requirement associated with it.

2.2 Functional Analysis

Shown in Figure 2, Functional Analysis and Allocation, is the iterative process of breaking down top-level project deliverables or functions into successive levels of sub-functions.

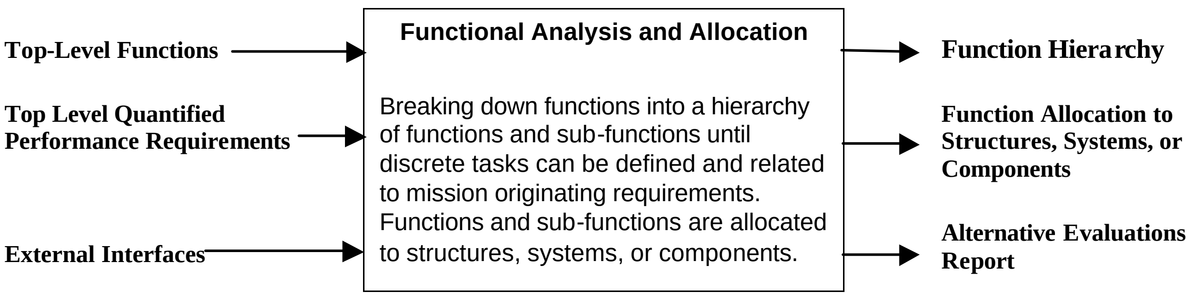

Figure 2. Functional Analysis and Allocation

This analysis is performed to develop the sub-functions that are necessary and sufficient to accomplish the top-level functions. These sub-functions form the key input to the work breakdown structure.

The work breakdown structure is a delivery-oriented grouping of all project elements (technical and non-technical) presented in graphic display to organize and subdivide the total scope of the project. The top-level set of functions are defined and documented during the pre-acquisition design phase. As the activity progresses through the conceptual and the preliminary design phases, initial functions are decomposed at each level into sub-functions to identify all the necessary and sufficient sub-functions to accomplish the parent function. This is an iterative process occurring as more information becomes available, such as studies and design selection.

At each level (system, subsystem, and component), sub-functions are identified based on the functions, requirements, and resulting design decisions from the previous level. As the level of detail increases, the sub-functions are allocated to systems, subsystems and/or components.

For complex activities, a functional hierarchy diagram may be used to depict the breakdown of functions into sub-functions. Also, a functional flow block diagram may be generated to show the logical relationship of functions or sub-functions at the system or subsystem level. The functional flow diagram may be used as a tool to analyze and model system behavior. An allocation matrix can be used to document which system, subsystem, or component performs the function and sub-functions.

2.3 Requirements

In concert with functions, requirements are the next step in the systems engineering process. Every function will have at least one requirement, and accurate formulation of requirements leads to assurance that the end product will satisfy the mission need. Once defined, the requirements form the basis for many other planning elements including design criteria, specifications, work breakdown structure, cost estimates, alternative evaluations, and value management studies.

Requirements state how well an architecture is to perform a function in terms of quantity (how many or how much), quality (how well), coverage (how much area, how far), timeliness (how responsive, how frequent) or readiness (availability, operational readiness). However, there are several origins/types of requirements that include:

- Those imposed on the project from external sources (constraints) such as regulations, laws, and policies
- Those related to interfaces between functions or to external systems (interface requirements)
- Those derived from within the project that define quantitatively how the functions is to be performed (performance requirements) in support of the end product.

Unless specified otherwise, the term “requirements,” includes all of the above.

The main attribute of a requirement is that it must be measurable/verifiable and that it applies to the architecture. Otherwise consideration should be given to the fact it may not be a requirement.

The initial, high-level set of project requirements is established during pre-acquisition activities. However, they are not normally of sufficient detail to produce the end product. Additional detail is developed during the project’s Definition and Execution phases through an allocation process.

No predefined level of detail exists to which the requirements definition should be developed in any particular phase. However, the requirements from any particular phase should form the basis for the specification for the next phase. The allocation of requirements through the project phases could be similar to the following:

- *Mission Analysis*. Through a high-level evaluation of project requirements, provides sufficient information to develop more detailed requirements during the Definition and Execution phases of the project.
- *Definition Phase*. Takes the high-level requirements developed during pre-acquisition activities and uses requirements allocation to determine requirements to the system and subsystem level.
- *Execution Phase*. Requirements allocation is used to determine requirements to the component level based on those developed for the system and subsystem levels during the Definition phase.

For complex projects, a database as described in the Functions subsection should be established.

2.4 Requirements Analysis

Requirements analysis is conducted to identify the necessary and sufficient set of performance requirements, design constraints, and interface requirements for each function. Generally, the PD/PM would perform this analysis for each identified function.

Performance requirements state how well the solution should perform the function. Requirements may be imposed on the design solution (design constraints) or be related to interfaces between systems (interface requirements). Identified requirements may result from the functions and requirements from the next higher level, or derived from an alternative study. Regardless of the source, each requirement is to have a documented basis. The depth to which this type of analysis is conducted for a given project is based on project risk and complexity, consistent with the depth of functional analysis performed. It may be performed at each level for large projects or at the system level for small low-risk modifications.

The requirements analysis and allocation step is illustrated in Figure 3.

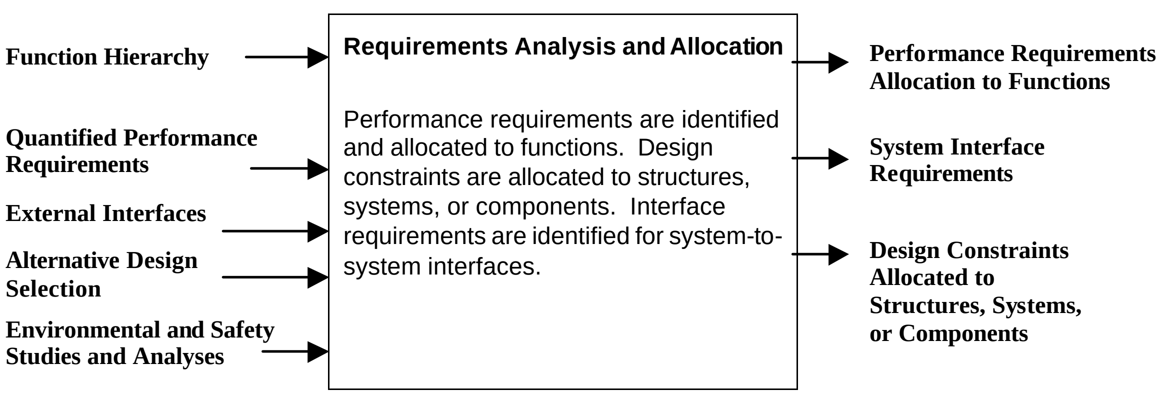

Figure 3. Requirements Analysis and Allocation

An initial top-level set of requirements is established and documented during the pre-acquisition design phase. Additional detailed requirements are developed during the conceptual and preliminary design phases.

Design constraints identified for systems are typically traceable to national codes and standards, DOE Orders, specific site standards, permits, Federal and state regulations and statutes, trade-off/alternative studies, and experience with similar systems. Because design constraints restrict the design of structures, systems, or components, they are allocated to structures, systems, or components rather than to functions.

The characteristics at the interface of two systems are used to identify and impose requirements (e.g., functional, performance, or physical constraints) on the interfacing systems. These are identified as interface requirements.

Value Management is a methodology for conducting alternative studies, inclusive of guidance on appropriate level-of-effort activities.

2.5 Architecture Identification

Architecture is the name given to the conceptualization of the end product at any level of allocation. The architecture at each level performs all functions and meets all requirements for that level. Usually, several different architectures can perform functions and meet requirements for a level of allocation, and these different architectures are called alternatives. The systems approach selects the preferred architecture from the alternatives at each level of allocation before proceeding to the next level.

The architecture, like the functions, and requirements, are initially very generalized or conceptual during the pre-acquisition activities. The objective should be to show the feasibility of the project under consideration. Risk areas should be identified with risk mitigation approaches establish as appropriate for high risk. If research and development is required, it can be initiated. Technology tradeoffs can be identified and made. The organization doing this work is usually a small focused core team.

The data being evaluated should be sufficiently complete in the cost and schedule area to allow management to make a Critical Decision-1. The baselined architecture at this point should be treated as a recommended approach for the Definition phase.

Functional allocation is not confined to any single phase of the life cycle and functional definition of the end product is not accomplished in any single life cycle phase. Successive phases of the life cycle develop the functional definition to lower, more detailed levels.

No predefined level of detail exists to which the functional definition should be developed in any particular phase. However, the functions from any particular phase should form the basis for the next phase. The allocation of functions in conjunction with the other steps through the project phases is an iterative process (see Section 10.1.2) and could be similar to the following:

- *Mission Analysis.* Through a high level evaluation of project functions, provides sufficient information to develop more detailed functions and requirements during the conceptual and execution phases of the project.
- *Definition Phase*. Takes the high-level functions developed during pre-acquisition activities and uses functional allocation to determine functions to the system and subsystem level.
- *Execution Phase*. Functional analysis is used to determine functions to the component level based on those developed for the system and subsystem levels during the Definition phase.

For complex projects, a database should be established to provide traceability of functions. The database should contain, at a minimum for all of the steps, the following information:

- The functions, requirements, enabling assumptions, and alternative evaluations by title including, for each, a description and a unique identifier;
- The functions, requirements, enabling assumptions, and alternative evaluations cross-referenced with one another and with any interfacing functions or requirements; and
- The functions cross-referenced with their parent function and child functions.

The cross-reference: (1) identifies the parent and child relationship for each function, (2) identifies requirements associated to each function, (3) identifies any enabling assumptions applicable to each function, and (4) identifies the alternative evaluations applicable to each function.

3.0 TYPICAL SYSTEMS ENGINEERING PROCESS

This section continues the discussion of the systems engineering processes by considering a typical systems engineering process from a *Project Management Overview* (see Figure 4). Figure 4 illustrates that for each phase of the life cycle it is necessary to:

- Define what must be done
- Determine how well actions/activities must be done
- Evaluate alternative solutions to completing actions/activities
- Select a solution

- Verify that the solution meets established requirements.

The systems engineering process shown in Figure 4 represents processes used by many organizations to define the system architecture for the end product. The process diagram may be entered at any of the four main elements in the upper part of Figure 4: requirements analysis, systems analysis and control, synthesis, and functional analysis allocation. Several iterations may be required before the final functions, requirements, and architecture are identified. This process diagram represents one part of the engineering process and should be used to develop requirements to the point that specifications can be prepared for the end product.

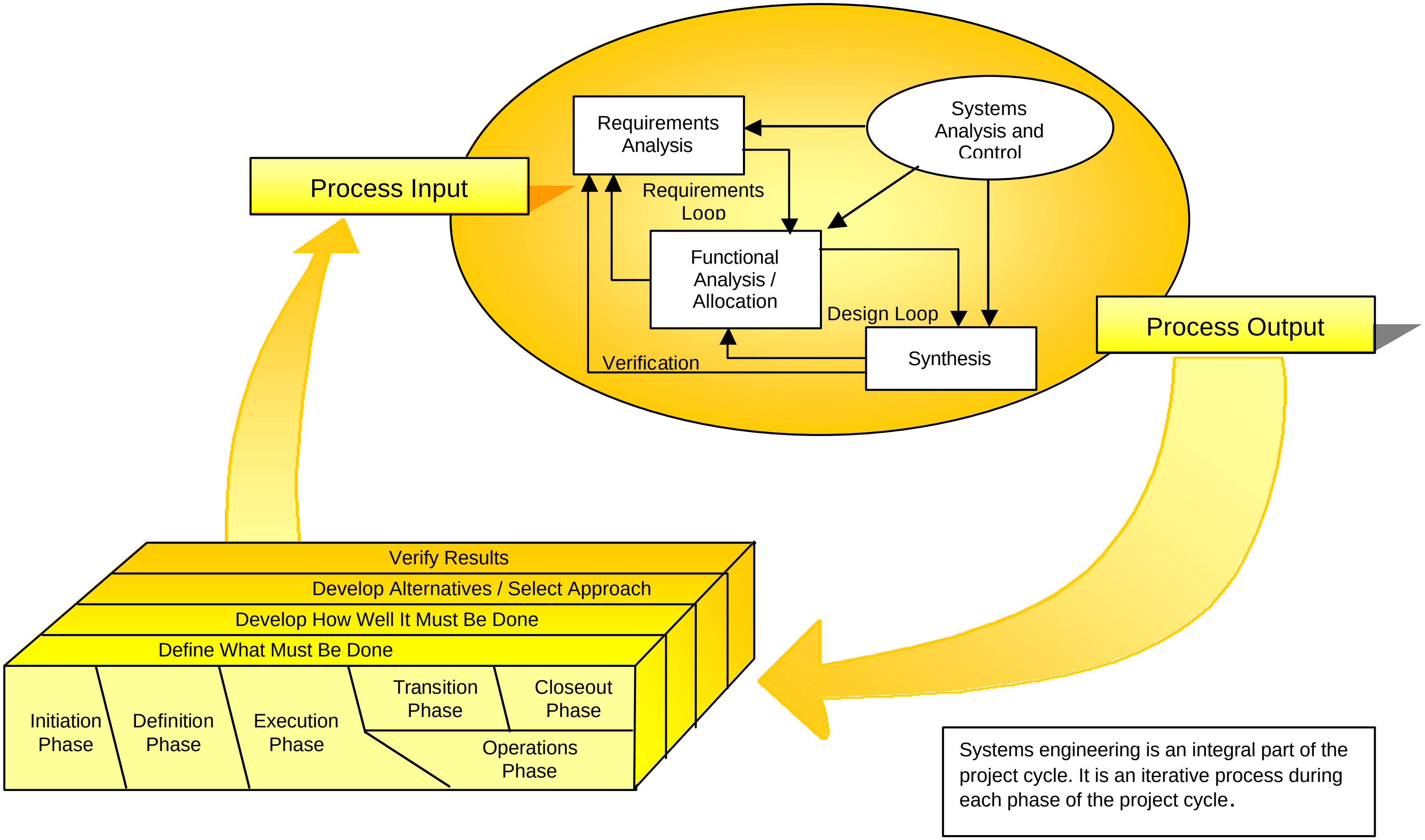

Figure 4. Typical Systems Engineering Process

3.1 Requirements Analysis

Requirements analysis establishes what the end product must do or be capable of accomplishing (functional requirements); how well the end product must perform in quantitative, measurable terms (performance requirements); the environment(s) in which the end product must operate; and constraints that affect design solutions. Requirements analysis is conducted iteratively with functional analysis to develop functional requirements and performance requirements to increasingly greater levels of detail. Functional requirements established in requirements analysis are used as the top-level functions for functional analysis.

3.2 Functional Analysis/Allocation

Functional analysis describes the problem in greater detail than that defined by requirements analysis. This is accomplished by breaking down (decomposition or allocation) the functions established in requirements analysis, to their sub-functions. Functional analysis/allocation is conducted iteratively with requirements analysis to accomplish the following:

- Define successively lower-level functions required to satisfy higher-level functional requirements.
- Define their functional interfaces and alternative architectures to meet the functional solutions.
- Identify architectural solutions for tradeoff studies.
- Define operational and environmental driven performance requirements in conjunction with requirements analysis and determine that higher level requirements are satisfied.
- Define flowdown performance requirements and design constraints
- Define and refine in conjunction with the synthesis process solution alternatives which meet requirements and allocate derived requirements to the physical architecture.

The extent of allocation depends upon establishing a clear understanding of what the end product must accomplish. Alternatives analyses and tradeoff studies are performed to select sub-functions that satisfy the previously established performance requirements.

3.3 Synthesis

A synthesis process translates the functional allocation into a physical architecture by grouping functions and sub-functions into logical physical elements that will make up the end product. These physical elements are composed of hardware, software, material, data, facilities, people, services, and/or processes. Alternatives are analyzed to determine which best satisfies allocated functional and performance requirements, derived requirements, interface requirements, and constraints. The synthesis process defines and integrates the physical architecture to a level of detail that enables verification that functional and performance requirements have been met. The process translates the architecture into a work breakdown structure, specifications, and configuration baselines.

3.4 Systems Analysis and Control

Systems analysis and control are used to measure progress, evaluate alternatives, select preferred alternatives, and document data and decisions. Systems analysis and control are discussed separately in the following paragraphs but are closely integrated in actual application.

- Systems analysis provides a rigorous quantitative basis for establishing requirements and physical architecture of the end product and assessing the effectiveness of the current solution. As part of system analysis, alternative analyses are performed to accomplish the following:
  - Resolve conflicts between functional and performance requirement and constraints that arise during requirements analyses.
  - Decompose functional requirements and allocate performance requirements during functional analyses.
  - Evaluate the effectiveness of alternative physical architectures and select the best solution during synthesis.
  - Identify and analyze risk factors.
  - Select appropriate risk handling approaches.
  - Manage risk factors.
- Control includes managing and documenting the activities of the engineering process, such as: (1) planning for the engineering process; (2) statusing activities and results which are used in other management and engineering process activities; (3) providing information for production, test, and operations; and (4) providing information for decision makers. Control mechanisms include risk management, configuration management, data management and performance based progress measurement, including reviews.

Material for the above discussion is taken from two commercial standards, EIA Standard IS-632 and IEEE P1220, which provide greater detail and are excellent references for the systems engineering process.

4.0 ALTERNATIVE ANALYSES AND TRADE-OFF STUDIES

Alternative analyses and trade-off studies are performed to identify and select sub-functions that satisfy established performance requirements. A synthesis process translates the functional allocation into a physical architecture by grouping functions and sub-functions into logical physical elements that will make up the end product. These physical elements are composed of hardware, software, material, data, facilities, people, services, and/or processes. Alternatives are analyzed to determine which best satisfies allocated functional and performance requirements, derived requirements, interface requirements, and constraints. A synthesis process defines and integrates physical architecture to a level of detail that enables verification that the functional and performance requirements have been met. The process translates the architecture into specifications, baselines, and a work breakdown structure.

4.1 Alternative Studies

As necessary, the PD/PM performs alternative studies and provides recommendations as to which of the alternate "methods" should be included in a given project or activity. An alternative study can be simple, ranging from one requiring little effort and minimal documentation to one requiring time, effort, and extensive reporting. The less the consequences of selecting one alternative over another, the less justification needed (See Figure 5).

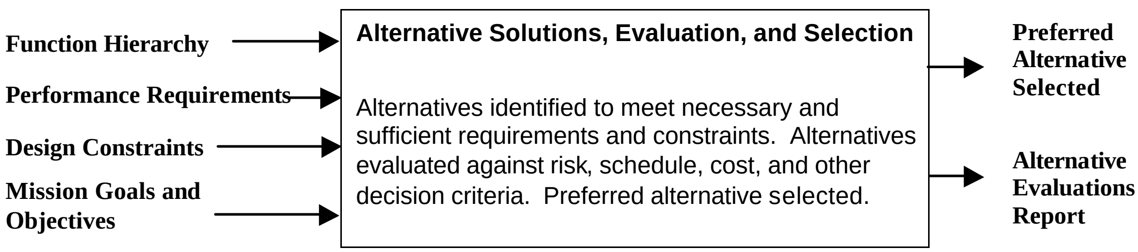

Figure 5. Alternative Solutions, Evaluation, and Selection

The use of alternative studies is an integral part of the project planning process. Alternative studies use the defined scope to establish a validated design.

Alternative studies are tailored to the intended application, inclusive of timing within the framework of the activity. Alternative studies are applied at all stages of the design process and may even be applied at the individual system or component levels. At the initial project level, the study should fully consider all life cycle cost impacts as input into making final decisions concerning which alternative to pursue.

The alternative study process is, at a minimum, a three-step process:

- Delineate the deliverables or function and requirements that are required.
- Identify alternatives that can meet the deliverables or function and requirements.
- Evaluate and select one of the alternatives.

For slightly more complex decisions, these steps are augmented with the establishment of criteria to be used in making the alternative selection. These criteria are selected to be independent of each other, and provide a distinction among the alternatives. Criteria are selected based on goals, objectives, and requirements. For highly complex applications, criteria are weighted to establish their relative importance in the decision-making process. Alternative studies are generally reported as a section of the PEP or as a separate Alternative Study Report.

4.1.1 Value Management

A value management (VM) study is a "special case" alternative study. This study (generally based on cost) evaluates whether there is a "better" value approach than the approach currently specified. Thus, VM requires a selected technical approach. The PD/PM should employ VM studies where appropriate, particularly to differentiate between repair, upgrade, new capability, etc.

A VM study is not synonymous with a cost cutting activity. However, this conclusion is a natural observation about any activity geared toward a "better" approach than that currently baselined—"better" usually meaning less costly. The term baseline is not limited to the traditional definition of having firm commitments (i.e., a Critical Decision-2 approval). Instead, baseline means that a preferred choice is identified and is being used as a basis for ongoing project work. By contrast, at the same stage of design, an alternative study has not yet selected a preferred approach (or the preferred approach has been determined to be unsuitable).

One distinction between VM and other alternative studies is that alternative studies are focused at a specific project level and often impose restraints on the alternatives to be considered. In VM, the life cycle of an activity (e.g., mission definition through decommissioning) is to be considered. The only constraints that should be considered in VM are those identifying the starting point for the study scope.

Consistent with the concept that VM is based upon identifying alternatives to the current approach, the optimum timing for use of VM is between conceptual and preliminary design. Earlier than this, there is usually insufficient detail to enable a full-scope evaluation (e.g., from mission identification through decommissioning). Later than this, there is often such an extensive investment in the selected approach that even highly valuable alternatives have insufficient benefit due to the effort required to implement the change, as opposed to maintaining the approach "as is." The "Value Management Guide" provides a complete description of the methodology for conducting alternative studies and VM studies, inclusive of guidance on appropriate level-of-effort activities.

4.2 Design Validation and Verification

Design validation and verification activities are conducted during the project life cycle to ensure that a project meets the defined mission by fulfilling the identified functions and requirements. Validation focuses strictly on the requirements and ensures the right problem has been defined. Verification focuses on the design solution for the validated requirements and ensures that the problem is solved. The formality and rigor of conducting and documenting validation and verification activities are commensurate with the associated risk, and the complexity of structures, systems and equipment.

4.2.1 Design Validation

The PD/PM should validate that the developed requirements are complete and technically adequate with respect to the mission, and are consistent with each successively developed level of detailed requirements. This is done by assuring that requirements flow down from the facility to systems, to appropriate components, and finally to a task. Validation includes a completeness check of the system-to-system interface requirements, and ensures that these interface requirements link to appropriate interface control documentation where necessary.

4.2.2 Design Verification

The PD/PM verifies that selected solutions meet validated requirements for high-risk structures, systems, and components. Verification methods that can be used include analysis, design reviews, system and component testing, and proof-of-principle demonstration testing. The rigor of verification performed is based on risk. Selected high-risk components could undergo several design reviews as well as testing to mitigate uncertainties associated with the identified risk. Also, proof-of-principle design demonstrations can be used for selected high-risk items where the test results help to define requirements and refine the final design.

4.2.3 Post-Construction Verification

After construction, the PD/PM should test and inspect systems in accordance with validated requirements and developed functional acceptance criteria. Post-construction verification ensures that functions and requirements are fulfilled; installed components function together as intended; and working systems perform as prescribed by mission needs. Post-construction verification is usually accomplished through testing.

5.0 INTERFACE MANAGEMENT

Interface management includes identifying system interfaces, defining the requirements at each system interface boundary, and managing (controlling) the interfaces during all steps of the systems engineering process. The interfaces are identified at the beginning of the task or project and continually managed throughout the life of the task.

5.1 Purpose

Interface management activities are conducted to ensure the proper operation of systems that interface with one another. These activities will reduce unnecessary redesign resulting from incompatible system designs.

5.2 What is Interface Management?

Interface management is a set of activities integrated into the systems engineering process where:

- All facility and system interfaces are identified
- Necessary interface requirements are clearly and completely defined
- Interfacing systems are designed to the same requirements
- Incompatibility issues are identified and resolved
- Changes made in one area of the system are checked for compatibility with other associated areas.

Interface management is focused on both external and internal facility interfaces. Interfaces between systems located in different facilities are considered external facility interfaces. Interfaces between systems located within the same facility are considered internal facility interfaces.

5.3 When Should Interface Management be Performed?

The systems engineering process (Figure 1), consists of four steps that are performed in a logical sequence, supported by three additional process steps, including interface management, that are performed concurrently with each of the sequential steps. Interface management activities are performed at each of these layers to ensure proper design and operation of the interfacing systems, structures, or components. The need for interface management and control begins at the beginning of the task or project when the system boundary is being established. The subsequent specific interface control activities, when they are performed (project life cycle and layer), and the extent to which they are performed are addressed in Section 5.4, Methodology.

5.4 Methodology

Interface management and control activities are performed in conjunction with the other systems engineering process steps. Specific interface control activities are described below.

5.4.1 Define System Boundary and Interfacing Systems

At the beginning of a task or project, the boundary or extent of the system being developed or modified must be established. This boundary defines the common interface between systems. The interfacing systems are then identified as well as the inputs and outputs flowing to and from the systems across the interface boundary. The interfacing systems may be internal or external to the facility containing the system to be developed or modified. The boundary, interfacing systems, and inputs/outputs should be determined as soon as the need for a system is identified. This work would typically be conducted during the program planning, pre-acquisition design, or conceptual design stage of the project or task.

These interfaces are documented on an interface block diagram as described in Section 5.5. This type of diagram is very helpful in communicating primarily functional interface information to the task/project team and responsible interfacing system engineers to obtain early agreement on task scope, system boundary, and system inputs and outputs. Interface block diagrams can be included in the appropriate task requirements document such as the facility design description and system design description to define applicable facility and system interfaces.

Early in a task or project, the specific parameters and characteristics of the inputs and outputs are not yet known. The lack of this information should not delay the preparation of the interface block diagrams since the identification of the interfacing systems and associated inputs/outputs are essential to establishing the system boundary. Definition of the parameters and characteristics will aid in the preparation of the interface requirements discussed in Section 5.4.2.

For complex systems, interface block diagrams may also be useful in establishing and communicating subsystem boundaries and the interfaces between subsystems.

5.4.2 Define Interface Functions and Requirements

The uncertainty of how systems will operate together should be reduced through the early identification of interface requirements. Any assumptions and/or important operational considerations in the minds of the task/project team and/or responsible interface engineers should be translated into interface requirements to ensure that designs are completed properly. During the requirements analysis step of the systems engineering process the necessary and sufficient functions and requirements for each system interface are defined and documented. These interface functions and requirements are defined as follows:

- *Interface Function*. Specifies "what" the interfacing system must perform (i.e., task, activity or action). These items clarify functional responsibilities of the interfacing systems.

  An example of this requirement is: The HT-Evacuation System transfers high tritium extraction gases from the Product Evacuation System to the HT-Thermal Cycling Absorption Process System during normal operation.

- *Performance Interface Requirement*. Specifies “how well” a function must be performed (e.g., parameters such as pressure, temperature, flow rate, gas composition, batch amount, transfer frequency, voltage, power, purity, water quality, etc.)

  An example of this requirement is: The evacuation pressure shall be <300 microns.

- *Design Interface Constraint*. Specifies codes and standards applicable to the interface, specific design, operating or maintenance configuration and essential features, materials, etc.

- *Physical Interface Requirements.* These specify physically related characteristics for components at the interface boundary such as materials, dimensions, tolerances, finish size, weights, dynamic limits, equipment envelopes, footprints, layout locations, orientation to plant reference marks, etc.

Functions and requirements are defined for each system-to-system interface on the interface block diagram and are documented in the system design description. These requirements should be established as early in the task or project as possible, but certainly not later than the preliminary design stage and/or procurement of long-lead engineered material. Responsible engineers for interfacing systems are identified by management and concurrence is obtained for the identified requirements via system design description review and approval. Concurrence by responsible interfacing system engineers is also necessary to change a requirement.

A distinction should be made between facility level requirements that flow down directly to the systems and the interface requirements that are imposed by one system upon another system to assure design compatibility at the interface. A common set of requirements such as plant transients (e.g., pressure, temperature, seismic, etc.) is often established by facility level analysis. These requirements are recorded in the system design description performance requirements section. These requirements are not appropriate as system-to-system interface requirements.

When system development and modifications are complex, interface control working groups may be necessary to develop interface agreements. Within these working groups, representatives develop documentation to establish the requirements to which the interface is designed and developed.

The interface control documents should not include a lot of detailed data that would normally be included on design drawings. The emphasis should be on identifying important parameters, characteristics, or configurations that need to be established to reduce risk and increase the probability that the interfacing systems will operate as desired.

5.4.3 Control Interfaces During Design

The interface control activities described below are conducted during the design phase of the task or project.

The alternative designs are checked to ensure that interface requirements (as well as performance requirements and constraints) are satisfied. If any alternative design does not meet an interface requirement, the alternative design must be eliminated from further consideration, the alternative must be revised to comply, or the interface requirement must be revised to permit compliance.

During the design of the selected alternative, design drawings such as piping and instrumentation diagrams are prepared to clearly show system-to-system interfaces. For complex systems, the interfaces between subsystems may also be depicted on the design drawings. Physical interfaces may be evaluated using a 3-dimensional computer model.

The design drawings (including procured engineered material) are checked for compliance with the interface requirements at one or more points (percent design complete) during the design execution stage. Upper tier and detail design documents and physical models are also checked to ensure that interfaces between systems and interfaces within the system, between subsystems and/or components, are functionally and physically compatible.

5.4.4 Verify Interfaces

Functional acceptance criteria and resultant test procedures are prepared based on developed requirements including the interface requirements. Testing is conducted in accordance with these test procedures to verify proper operation across the interfaces.

5.5 Interface Block Diagram

The Interface block diagram is used to provide a graphical representation of the system boundary, the interfaces with other systems and the input and outputs flowing between the systems. This diagram, in conjunction with text descriptions, can provide an adequate functional definition of the system interfaces, especially early in the task or project.

The diagram consists of solid blocks with labeled names that represent systems and their functional and physical boundaries. The boundary for each system is drawn to enclose all subsystems and components within the system that are performing either a system, subsystem, or component function. Arrows are drawn between the blocks to represent the medium flowing between the system boundaries. The arrows are positioned to show the direction of flow between the systems, and they are tagged with a short descriptive label. All functional and physical interfaces should be represented. If all interfacing systems are not included, this condition should be so noted on the diagram. (For example, for clarity it may be practical to show only process systems on the diagram. In this case, the diagram should indicate that support system interfaces have been omitted and provide a reference to another diagram that shows these interfaces.)

5.6 Interface Control Document

The interface control document is a management tool formalizing an agreement between two or more systems that functionally and physically connect. The interface control document provides the functional, physical, and performance requirements and design constraints for one or more system-to-system interfaces. The interface control document is prepared in the form of either a drawing or document.

Interface control documents should not include a lot of detailed data that would normally be included on design drawings. The emphasis should be on identifying important parameters, characteristics, or configurations that need to be established to reduce risk and increase the probability that the interfacing systems will operate as desired.

Typical information that would be addressed in an interface control document for the interfacing systems include:

- Document purpose
- Document scope
- Summary description of interface (block diagram may be included here to show system inputs and outputs and what is flowing across the interface.)
- Interface functions
- Performance interface requirements and basis
- Design interface constraints and basis
  - — Interface configuration
  - — Interface essential features
  - — Limitations
- Physical interface requirements
- Boundaries
  - —Project boundary
  - —System boundary
  - —Physical boundary
- References to design drawings and documents

This is an example of interface block diagrams at the facility and system levels.

Examples	Title	Page
#1 Figure E.A-1	Interface Block Diagram Example #1 TFM&C/233 –H Process Systems Gas Transfer Interfaces	E-16
#2 Figure E.A-2	Interface Block Diagram Example #2 Radioisotope Power Systems Production Facility Interfaces	E-17
#3 Figure E.A-3	Interface Block Diagram Example #3 Interface Diagram for Injector System	E-21

This page is intentionally blank.